

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

**Defective images within this document are accurate representations of the original documents submitted by the applicant.**

**Defects in the images may include (but are not limited to):**

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

Requested Patent: JP1068835A  
Title: SOFTWARE RIGHT MANAGEMENT CONTROL METHOD ;  
Abstracted Patent: JP1068835 ;  
Publication Date: 1989-03-14 ;  
Inventor(s): MORI RYOICHI; others: 01 ;  
Applicant(s): RYOICHI MORI ;  
Application Number: JP19870227106 19870910 ;  
Priority Number(s): ;  
IPC Classification: G06F9/06 ;  
Equivalents: JP2723231B2 ;

**ABSTRACT:**

**PURPOSE:** To perform the right management superior in maintainability and flexibility in the environment of multiprogramming by individually assigning the key, with which ciphered instructions are deciphered, to a program or a program group as the right management object and switching the deciphering keys by a right management interrupt.

**CONSTITUTION:** A right management interrupt generating mechanism 13 which controls the change related to the right management state is provided independently of the supervisor interrupt which starts the processing function of an operating system or the like. When the right management interrupt is generated by the right management interrupt generating mechanism 13, the right management state is saved to a right management table 17 and a current right management ID 16 of a program status word 15 is changed by a right management state change processing 14, and the key with which instructions are deciphered is switched. Thus, right management in the environment of multiprogramming is possible.

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭64-68835

⑬ Int.Cl.<sup>4</sup>

識別記号

庁内整理番号

⑭ 公開 昭和64年(1989)3月14日

G 06 F 9/06

3 3 0

A-7361-5B

審査請求 未請求 発明の数 1 (全13頁)

⑮ 発明の名称 ソフトウェア権利管理制御方法

⑯ 特 願 昭62-227106

⑰ 出 願 昭62(1987)9月10日

⑱ 発 明 者 森 亮 一 東京都文京区白山1-24-12  
⑲ 発 明 者 田 代 秀 一 千葉県柏市松葉町1-12-20-2  
⑳ 出 願 人 森 亮 一 東京都文京区白山1-24-12  
㉑ 代 理 人 弁理士 小笠原 吉義 外2名

明 細 書

1. 発明の名称 ソフトウェア権利管理制御方法

2. 特許請求の範囲

(1) 暗号化された命令を、命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって、

権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構(13)を設け、

権利管理対象となるプログラムまたはプログラム群に対し、暗号化された命令を復号化する鍵を個別に割り当て、

上記権利管理割込みにより、上記復号化する鍵を切り換える(14)ことを特徴とするソフトウェア権利管理制御方法。

(2) 上記権利管理対象となるプログラムまたはプログラム群は、許諾条件プログラム、該許諾条件プログラムによって実行を制御されるソフトウェア本体、またはそれらの組み合わせであること

を特徴とする特許請求の範囲第(1)項記載のソフトウェア権利管理制御方法。

3. 発明の詳細な説明

(概要)

個々の権利対象ソフトウェア毎に、ソフトウェアの使用許諾、使用記録情報の収集などを、他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関し、

マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的とし、

暗号化された命令を、命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって、権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構を設け、権利管理対象となるプログラムまたはプログラム群に対し、暗号化された命令を復号化する鍵を個別に割り当て、上記権利管理割込み

により、上記復号化する鍵を切り換えるように構成する。

#### (産業上の利用分野)

本発明は、個々の権利対象ソフトウェア毎に、ソフトウェアの使用許諾、使用記録情報の収集などを、他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関する。

半導体技術の進歩によって、計算機ハードウェアの価格低下、小型化、高機能化が急速に進みつつある。ハードウェア機能の高度化は、ソフトウェアの高度化を要求し、結果として、ハードウェア開発者、ソフトウェア開発者、利用者の明確な分業化が促進されている。

しかしながら、ソフトウェア流通の環境は、現在、混乱した状況にあるといつてよく、その環境を改善するために、ソフトウェア権利者の保護と利用者の便利さとを両立させるシステムであつて、

- 3 -

上記③は、バックアップがとれなくなるという問題や、複製不能にしている手段を調べ、それを回避することにより複製可能となるという問題があり、上記④は、特定計算機を識別するために、ソフトウェアの販売時に特殊化処理が必要となり、自由なソフトウェアの流通が阻害されるという問題がある。

これらの問題を解決するものとして、本出願人は、ソフトウェア権利者の権利と、利用者の自由と、流通の容易性の3つを同時に満たすソフトウェアサービスシステム(SSS)の基本構想を提案している。〔発明一: "ソフトウェアサービスシステム", 電子通信学会誌, Vol.67, No.4, pp.431-436 (Apr.1984)〕

ここで提案されたソフトウェアサービスシステムの基本は、

- (1) ソフトウェアは、いかなる条件が成立した場合に、その実行を許すかの許諾条件を内部に持つべきである。
- (2) 計算機内部には、利用者の持つ権利を記述し

ソフトウェアの自由な大量流通を支援するシステムを構築することが望まれている。

#### (従来の技術)

ソフトウェアの開発には、大量の人員および時間を要するにもかかわらず、利用者または第三者は、完成したソフトウェアを複製して、全く同じものを作成することが比較的容易にできる。そのため、ソフトウェアを開発した権利者(ソフトウェア権利者)は、開発コストの回収が不能になるケースが多く、これに対して、(1)ソフトウェア製品の価格を盗用による損失分を上乗せした値とする、(2)契約によって無断複製を禁止する、(3)ソフトウェアを複製不能な媒体に封入して流通させる、(4)計算機のシリアルナンバーなどにより使用可能な装置を限定する、などによって対処することが行われている。

しかしながら、上記(1)は、正当な利用者にも過度の負担を与え、上記(2)は、複雑な契約手続きが必要になると共に、それによる効果が充分でなく、

- 4 -

た権利記録があるべきである。これには、共通クレジット、買い取り記録、試用記録、特別許諾コード等が有り得る。

(3) 計算機は、上記④の権利記録が上記(1)の許諾条件を満足した場合にのみ、ソフトウェアの利用を許す実行管理機構を持つべきである。

(4) 権利者が、ソフトウェアの利用状況を把握できるように、計算機は、回収作業記録を持つべきである。利用者に特別な面倒を与えることなく、この作業記録を回収する電子的手段が有り得る。…というものである。

ところで、このソフトウェアサービスシステムの提案とは別に、暗号化された命令およびデータを主記憶にロードし、命令フェッチ時に、計算機のシリアルナンバーなどの復号鍵によって復号して実行する計算機アーキテクチャが知られている。

#### (発明が解決しようとする問題点)

上記ソフトウェアサービスシステムを、マルチプログラミングの環境下で実現する場合、複数の

ソフトウェア権利者に係る複数のプログラムが、時分割的に同時に計算機資源を利用して、走行することになる。この場合、第1に、権利の管理に係るハードウェア機構を制御する主体を、いかにして許諾条件をチェックするプログラム（以下、許諾条件プログラムという）に限定するか、第2に、マルチプログラミングの環境において、許諾条件プログラムと、それによって実行を制御されるソフトウェア本体との対応関係をいかにして保つか、第3に、オペレーティング・システムに対し、スワッピングやプロセス・スイッチを行うためのアクセス権を認めつつ、許諾条件プログラムの内容を改ざんしたり、権利管理のための流れを乱したりすることがないように、いかにしてアクセス権を制限するか、といったことに対する考慮が重要となる。従来、このような権利管理に対する適切な計算機の実行制御を行う機構はなかった。

本発明は上記問題点の解決を図り、マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的としている。

- 7 -

件プログラム11とソフトウェア本体12との対によって構成される。これらの全部または一部は、権利管理対象プログラム10毎に定められた1または複数の暗号鍵によって予め暗号化されている。

権利管理対象プログラム10のロード時に、暗号化された命令を復号化する鍵情報が、その権利管理対象プログラム10に対してシステムでユニークに割り当てた権利管理ID16と共に、権利管理テーブル17に書き込まれる。

プログラムステータスワード15は、カレントの権利管理ID16が設定されるフィールドを持っている。プログラム実行時に、プログラムステータスワード15に設定されている権利管理ID16により、権利管理テーブル17の該当エントリがアクセスされ、そのエントリ中の復号鍵(KEY)が読み出されて、復号化回路18へ送られる。復号化回路18は、暗号化された命令をフェッチしたときに、権利管理テーブル17から読み出した復号鍵によって復号し、その復号した命令を計算機の命令実行ユニットへ送る。

- 9 -

（問題点を解決するための手段）

第1図は本発明の原理説明図である。

第1図において、10は権利管理の単位となる権利管理対象プログラム、11は許諾条件をチェックしソフトウェア本体の実行を制御する許諾条件プログラム、12はワープロソフト、コンパイラなどの利用者の使用目的に応じた処理を行うソフトウェア本体、13は権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構、14は権利管理割込みによって権利管理状態を変更する権利管理状態の変更処理、15はプログラムステータスワード(PSW)、16は権利管理対象プログラム10を識別する権利管理ID(PMID)、17は権利管理対象プログラム10の暗号化された命令を復号化する鍵を権利管理ID16毎に記憶管理する権利管理テーブル、18は暗号化された命令をその命令のフェッチ時に復号化する復号化回路を表す。

権利管理対象プログラム10は、例えば許諾条

- 8 -

本発明では、オペレーティング・システム等の処理機能を起動するスーパーバイザ割込みなどとは別に、権利管理状態に関する変更を制御する権利管理割込み発生機構13を設ける。そして、権利管理割込み発生機構13により、権利管理割込みが発生した場合に、権利管理状態の変更処理14によって、権利管理テーブル17への権利管理状態の退避やプログラムステータスワード15のカレントの権利管理ID16の変更などを行う。これにより、命令などを復号化する鍵を切り換える。

（作用）

本発明によれば、割込みが、オペレーティング・システムによる資源管理などに関連する従来の割込みと、権利管理状態を制御するための権利管理割込みの2系統に分離されることになる。

権利管理割込みが発生したときに、権利管理状態の変更処理14によって、権利管理テーブル17に基づく権利管理状態を変更するので、権利管理に係るハードウェア機構を制御することができ

- 10 -

る主体を、特定の許諾条件プログラム11などに個別に限定することができる。また、個々に復号鍵が管理されるので、ある許諾条件プログラム11に対応した権利管理テーブル17中のエントリ等、権利管理に関する情報を、他の権利管理対象プログラム10などによって、改ざんされることを防止することができる。

また、割込みが、少なくとも従来の割込みと権利管理割込みの2系統になるので、オペレーティング・システムによる権利管理に対する望ましくない関与も、防止することが可能となる。

#### (実施例)

第2図は本発明を用いたソフトウェアの流通形態の例を説明するための図、第3図は本発明による権利管理の説明図、第4図は権利管理状態の状態遷移図、第5図は本発明に用いるハードウェア構成例、第6図は権利管理対象となるソフトウェアの構成例、第7図は権利管理テーブルの構成例、第8図はP割込み要求テーブルの構成例、第9図

はS割込み制御の例、第10図はS割込みからの復帰制御の例、第11図は本発明の一実施例におけるスタックの使用例、第12図は本発明の一実施例における空間と鍵の関係説明図を示す。

本発明は、ソフトウェアの保護と利用促進のため、例えば第2図に示すようなソフトウェアの流通形態を採用する場合に用いることができる。

① ソフトウェアを開発したソフトウェア権利者は、そのソフトウェアを流通させる場合、ソフトウェアの利用者に対して要求する条件を記述した許諾条件プログラムを作成し、それを開発したソフトウェア本体と結合して、暗号化し、電波による放送も含めた自由な媒体により、ソフトウェアを配布する。

② ソフトウェアを利用したい利用者は、例えば予め共通クレジット(CC)を自動販売機などのベンディングマシンにより購入する。この共通クレジットの内容は、例えばICカードによる媒体を介して、データ処理装置20に入力できるようになっている。

- 11 -

③ 利用者は、共通クレジットの内容が記録されたICカードを、本発明に係るソフトウェア権利管理制御機能を持つデータ処理装置20に装着し、適当な流通路から入手したソフトウェアを、データ処理装置20上で動作させる。これにより、そのソフトウェアの許諾条件プログラムが復号化されて動作し、ICカードへのアクセスにより、共通クレジット情報に関連するチェックなどを行う。必要に応じて共通クレジットに記録されている料金の減算を行う。チェックに合格した場合にのみ、必要なソフトウェア本体の実行を可能とする制御を行う。そのとき、その利用に関する作業記録情報を蓄積しておき、その作業記録(AR)を適当な時期にICカードへ転記する。

④ ICカードに書き込まれた作業記録は、例えば共通クレジットを継続使用するために、新たな料金の支払いにより共通クレジットの内容を再設定するとき、ベンディングマシンに読み取られ、収集される。この作業記録情報を参照することにより、ソフトウェア利用状況の統計をとり、それ

- 12 -

に基づいて複数のソフトウェア権利者間で妥当な料金の分配を行うことができる。

第2図に示したソフトウェア流通形態は、一例であり、本発明は、共通クレジットではなく、各ソフトウェア個別の利用資格情報を管理する流通形態をとる場合にも採用することができる。また、ICカード以外の媒体を用いることも可能である。

本発明による場合、データ処理装置20は、複数の独立したソフトウェアを、マルチプログラミングによって並列動作させることができる。

本発明による権利管理は、概念的には、第3図に示すようになる。

マルチプログラミングの環境では、オペレーティング・システムが、入出力装置の割り当て、プロセッサの割り当てなどの計算機ハードウェア資源の管理を行っている。この場合、誤ったプログラムの実行などによって支障をきたさないようにするために、また計算機の資源を多く利用しようとする利用者によって、資源が恣意的に用いられることを防ぐために、計算機の実行状態をユーザ

状態と、スーパーバイザ状態に分けることが広く行われている。

利用者のプログラムは、ユーザ状態で実行され、資源管理用のオペレーティング・システムは、通常、スーパーバイザ状態で実行される。そして、利用者による資源の濫用を防止し、また利用者が勝手に実行状態を変更できないようにするために、ユーザ状態からスーパーバイザ状態へと状態を変更する手段を、割込み（割出しを含む）に限ることによって、両状態の分離を図っている。

ここで、第2図で説明したような許諾条件プログラムの復号化や、共通クレジットの読み書きなどに関連する権利管理を、スーパーバイザ状態のもとで、オペレーティング・システムにより制御するとすれば、利用者が自己の都合のよいように内部を改変したオペレーティング・システムをロードすることに関して、抵抗力を持たない。これに対し、オペレーティング・システムをファームウェア化して、内容の変更を困難にするという対応策も考えられるが、オペレーティング・システム

のバージョンアップが不可能になるなどの他の問題が発生する。

そこで本発明では、スーパーバイザ状態とは異なる権利管理状態という特別な実行状態を新設している。権利管理状態は、ソフトウェアをできるだけ無料で実行したいと考える利用者があり得る環境の中で、利用者の故意または過失による介入を避けつつ、外部から供給されたソフトウェアの権利に関する管理を行うためのプログラムを実行可能とする状態である。

権利管理状態は、他の状態からそこへ状態を変更する手段を割込みに限っている点および他の状態では実行できない命令を持つ点で、スーパーバイザ状態に類似しているが、暗号の応用と鍵の管理とによって不正なアクセスの防止を図る点が、スーパーバイザ状態と大きく異なる。

第3図に示すように、オペレーティング・システムが、スーパーバイザ状態のもとで、資源管理を行うのに対し、権利管理対象となっているプログラムの各許諾条件プログラム11A、11B、…

- 15 -

は、個々の権利管理状態のもとで、ソフトウェア本体12A、12B、…の実行を監視することによって、権利管理を行う。以下、許諾条件プログラムと、それによって制御されるソフトウェア本体とを合わせて権利管理対象という。

権利管理状態を設けることにより、本発明に係るデータ処理装置は、例えば第4図に示すような状態遷移を行う。ユーザ（US）状態、スーパーバイザ（SV）状態、権利管理（PM）状態があり、スーパーバイザ状態は、さらにSVp状態とSVu状態とに分かれる。SVp状態とSVu状態とは、使用できる命令、アクセス権等に関しては全く同様であるが、割込みからの復帰命令を実行した場合に、PM状態に復帰するかUS状態に復帰するかの点が異なる。

以上の4つの状態（SVp、SVu、PM、US）は、独立した2つの状態ビットにより表現される。その1つは、プログラムステータスワード15中に置くSビットであり、他の1つは権利管理テーブル17中に置くPビットである。

- 17 -

- 16 -

Sビットの変化に伴う状態遷移は、スーパーバイザ状態で実行するプログラム、即ち、オペレーティング・システムの制御に関係するものであり、Pビットの変化に伴う状態遷移は、PM状態で実行されるプログラム、即ち、許諾条件プログラムの制御に関係するものである。

これら2系統の状態遷移を直交した関係におき、互いに他の系統の状態遷移には直接影響を及ぼさないようにしているので、資源管理と権利管理との制御機構の分離が図られている。

以下、Sビットに関連する従来の割込みをS割込みといい、Pビットに関連する本発明に係る権利管理割込みをP割込みという。

本発明に用いるハードウェア構成は、例えば第5図に示すようになっている。

第5図において、第1図と同符号のものは、第1図に示すものに対応する。30は命令実行ユニットである演算/制御、31はICカードインタフェース、32はICカードから入力された共通クレジット情報を記憶する共通クレジットレジ

- 219 -

- 18 -

スタ(CCR)、33はソフトウェア買い取り情報などの個々のソフトウェアに特有な権利管理情報を持つ個別権利メモリ(SRM)、34はソフトウェアの利用実績情報が記録される作業記録メモリ(ARM)である。

共通クレジットレジスタ32、個別権利メモリ33、作業記録メモリ34は、不揮発メモリによって構成される。

35は許諾条件プログラムまたはソフトウェア本体の全部または一部をDES方式により復号化するDES暗号機構、36はP割込みの要因などの制御情報を記憶するP割込み要求テーブル(PIT)、37は権利管理対象となるソフトウェアをロードする際に後述するキー格納部の復号化を行う公開鍵暗号機構である。

第5図に示す装置が扱うソフトウェアの構造は、例えば第6図に示すようになっている。キー格納部、許諾条件プログラム11、ソフトウェア本体12の3つの部分からなる。

本実施例では、許諾条件プログラム11を、権

利者が任意に与える鍵(KEY1)によって、いわゆるDES方式で暗号化する。また、ソフトウェア本体12の少なくとも一部を、やはり権利者が任意に与える鍵(KEY2)によってDES方式で暗号化する。なお、DES方式については、例えば「一松信監修：『データ保護と暗号化の研究』、日本経済新聞社〔1983〕」に記載されている。

キー格納部には、このソフトウェアを識別するユニークな権利番号(PN)と、個別権利記録(SR)に対するアクセスを管理するための非公開のSRアクセスキーと、許諾条件プログラム11、ソフトウェア本体12を復号化するためのKEY1、KEY2とが設定されるようになっている。キー格納部は、例えば公開鍵暗号方式のひとつであるRSA法により暗号化される。なお、このRSA法も、例えば上記「データ保護と暗号化の研究」の著書などにより知られている。公開鍵暗号方式は、暗号化に用いる鍵を知っていても、秘密鍵である復号化鍵を知っていないと、暗号を

- 19 -

解くことができないという特徴がある。もちろん他の暗号方式を用いても、同様に本発明を実施することは可能である。

このキー格納部を復号化するための鍵は、ハードウェア製造時に、第5図に示す公開鍵暗号機構37の内部に封入しておく。このソフトウェアを主記憶上にロードする際に、公開鍵暗号機構37によってキー格納部を復号化し、その復号結果を権利管理テーブル17に設定する。

権利管理テーブル17は、例えば第7図に示すような権利管理に関する情報を持つ。

権利管理テーブル17中のレコードは、このレコードに対応する権利管理対象を識別するためのPMID、権利番号(PN)、SRアクセスキー、許諾条件プログラムを実行するために必要なDES鍵(KEY1)、ソフトウェア本体を実行するために必要なDES鍵(KEY2)、KEY2の有効/無効を示すビット(K2F)、許諾条件プログラムによる権利管理状態か否かを表すPビット、対応する権利管理対象がロードされた直後であ

- 20 -

ることを表すイニシャルビット、およびスタックの正当性を確認するためのスタックチェックコードの9つのフィールドからなる。

システム中に同時に存在し得る権利管理対象数は、権利管理テーブル17のレコード数に等しい。複数のレコードのうち、PSW中のカレントPMIDの値と一致したPMIDを持つレコードを、カレントPMTレコードと呼び、権利管理テーブル17のレコードに対するアクセスは、通常の場合、このカレントPMTレコードに対して行われる。

第5図に示すP割込み要求テーブル(PIT)36は、権利管理割込み(P割込み)発生機構13が使用するテーブルであり、例えば第8図に示すような構成になっている。PIT36中の各レコードは、P割込みを要求する権利管理対象のPMID、そのSRアクセスキーを記すフィールド、要求するP割込みの要因を記述するフィールド(P-Reason)、P割込み発生時の飛び先アドレスを記述するフィールド(New-PC)、P割込み



要求の要因に対するパラメータを記述するフィールドの5つのフィールドからなる。

PIT36中のレコードは、PM状態でしか使用できない。権命令によってのみ、生成/消去が可能にされ、また、SRアクセスキーを用いたアクセス権のチェックにより、正当な権利者以外の許諾条件プログラムからは、既に存在するPITレコードの書き換えや消去ができないようになっている。

P割込みの要因には、例えばUS状態におけるP割込み命令の実行、タイマー割込み、US状態における特定命令の実行(命令トラップ)がある。タイマー割込みに対しては、パラメータとして割込み発生時の時間間隔を与えることができ、命令トラップに対しては、任意の命令コードをパラメータとして与えることができる。これにより、許諾条件プログラムは、必要に応じてソフトウェア本体の実行制御を行い、またソフトウェア本体の実行に関する作業記録をとる契機を得ることができるようになっている。

- 23 -

- トPMTレコードから取り出した旧スタックチェックコード1, 2の4ワード(64bit)をKEY1で暗号化し、スタックへストアする。
- (a) 新スタックチェックコード1, 2を、カレントPMTレコードへストアする。
  - (f) PITの飛び先アドレス(New-PC)を、プログラムカウンタにセットする。そして、(d)の制御へ移る。
  - (d) Pビットが0の場合には、割込みベクタをプログラムカウンタにセットする。
  - (a) PSW、プログラムカウンタ、スタックフォーマットコードをスタックへストアする。
  - (i) PSWのSビットを"1"にし、オペレーティング・システムの割込み処理ルーチンへ制御を移す。

なお、この例では、S割込み時における飛び先アドレス、即ち、プログラムカウンタに設定する新しい値を、いわゆる割込みベクタテーブルからロードする方式を用いている。

スーパーバイザ状態において、RSB(Return

第5図に示すP割込み発生機構13は、許諾条件プログラムがPIT36に指定した要因が発生したときに、自動的にP割込みを起こす機構であるが、その内部の回路構成については、従来技術と同様な一般的な割込み技術を用いて実施可能であるので、これ以上の説明を省略する。

S割込みとP割込みとを独立させるために、S割込みが発生した場合におけるプロセッサの動作は、例えば第9図に示す(a)~(i)のようになる。

- (a) S割込みが発生した場合、ファームウェアなどにより、カレントPMTレコードのPビットの0/1をチェックし、0であれば(a)へ制御を移す。
- (a) Pビットが1であれば、現在のPSW及びプログラムカウンタの値から、16bitのサイクリックコードを生成し、新スタックチェックコード1とする。
- (b) 16bitの乱数を生成し、新スタックチェックコード2とする。
- (d) 新スタックチェックコード1, 2及びカレン

- 24 -

from S-Exception) 命令を実行することにより、PM状態またはUS状態のいずれかへ遷移する。そのときのプロセッサの動作は、例えば第10図に示す(a)~(d)のようになる。

- (a) カレントPMTレコードのPビットの0/1をチェックする。それが0であれば、(f)へ制御を移す。
- (b) スタックから、スタックチェックコード1, 2および旧スタックチェックコード1, 2を取り出し、それらをKEY1で復号化してプロセッサ内に保持する。
- (a) スタックのPSW及びプログラムカウンタの値から、16bitのサイクリックコードを生成し、上記(a)で取り出したスタックチェックコード1と比較する。不一致であれば、エラーとして処理する。即ち、PM状態への遷移などを防止する。
- (d) 次に、カレントPMTレコードのスタックチェックコード2と、上記(a)で取り出したスタックチェックコード2とを比較する。不一致であ

ればエラーとして処理する。

- (a) 旧スタックチェックコード1, 2をカレントPMTレコードへストアする。
- (f) PSWのSビットを"0"に戻す。
- (d) スタックにストアされているプログラムカウンタの値を、プログラムカウンタにセットする。そして、割込み発生直前に実行されていたルーチンへ復帰する。

PM状態からS割込みによりSV<sub>0</sub>状態へ遷移する際に、例えば第11図(a)に示すようなスタック情報が、プロセッサによってスーパーバイザ空間に遷送される。スタックチェックコードは、スタックに遷送された情報が、オペレーティング・システムによって書き換えられることを防止するために使用されるコードである。

スタックチェックコード1は、第9図で説明したように、スタックに遷送されるPSW及びプログラムカウンタの値から生成するサイクリックコードであり、スタックチェックコード2は、16bitの乱数である。これらは、スタックに遷送さ

- 27 -

PM状態におけるプログラムの実行は、以下のとおりである。

PM状態において、命令フェッチ、データのロード/ストア、サブルーチン呼び出し、割込み発生時におけるプログラムカウンタのスタックへの遷送など、主記憶に対するあらゆる書き込み、読み出しの際に、カレントPMTレコード中のKEY1を用いた自動的な暗号化および復号化が、第5図に示すDES暗号機構35によって行われる。PM状態で使用するアドレス空間は、US状態の空間と全く同一であり、区別されていない。これらは、鍵の管理によってのみ隔離されている。

オペレーティング・システムといえども、鍵を知らない限り、許諾条件プログラムを一時停止、強制終了させること以外、その動作に不正に介入することはできない。

PM状態からUS状態への戻りは、RPE (Return from P-Exception) 命令の実行によって行われる。

US状態においては、命令フェッチ時に限り、

れると同時に、カレントPMTレコードにも格納され、後にスタックの情報を取り出す際にチェックされる。これにより、スタック内容が書き換えられたことを検出できるようになっている。

旧スタックチェックコード1, 2は、割込みがネストされた場合に、古いスタックチェックコードを記録するためのものである。

US状態からSV<sub>0</sub>状態へ遷移する場合には、第11図(b)に示すデータがスタックに遷送される。また、SV状態時におけるS割込み発生時には、第11図(c)に示すデータがスタックに遷送され、US状態時におけるP割込み発生時には、第11図(d)に示すデータがスタックに遷送される。なお、SV状態においては、P割込みはマスクされる。

本実施例では、SV、PM、USの各状態毎に、プログラムのアクセスできる空間およびハードウェアによって自動的に使用される暗号の鍵が決められる。第12図は、その空間と鍵の関係を示している。

- 28 -

カレントPMTレコードのKEY2を用いた復号化を行う。ただし、復号化による命令実行速度の低下を軽減するために、プログラムのすべての部分を暗号化するのではなく、暗号化する範囲を任意に指定できるようにする方式をとることが望ましい。そのため、US状態で、カレントPMTレコードのK2Fビットをコントロールする命令が用意され、K2Fビットが"1"の間だけ、命令フェッチ時に、KEY2による復号化を自動的に行うようになっている。

オペレーティング・システムは、以下の処理を行う。

① 権利管理対象のソフトウェア全体を、暗号化されたまま主記憶へロードする。

② 次に、権利管理テーブル17内に新しいレコードを生成する命令を発行する。この命令では、これから実行しようとする権利管理対に与えるPMIDの値と、ロードしたソフトウェアのキー格納部の先頭論理アドレスとをパラメータとする。

PMIDの値は、システム内部においてユニーク

であれば、任意でよい。この命令によって、ハードウェアは、ソフトウェアのキー格納部を復号化し、その情報を権利管理テーブル17に格納して新しいPMTレコードを作成する。同時に、PSWのカレントPMIDのフィールドを、これから実行しようとする権利管理対のPMIDの値にセットする。権利管理テーブル17のイニシャルビットおよびPビットを"1"にセットする。

③ RSE命令を実行する。カレントPMTレコードは、新しく生成されたレコードであり、このイニシャルビットが"1"である場合に、RSE命令が実行されると、プロセッサの状態はPM状態に遷移し、許諾条件プログラムに実行制御が渡される。同時に、イニシャルビットはクリアされ、以後、オペレーティング・システムから許諾条件プログラムへ制御を移す手段は、S割込みに対するRSE命令の実行以外に存在しなくなる。

④ 上記③のままで、権利管理対が生成され、許諾条件プログラムの先頭から実行が開始される。許諾条件プログラムの先頭部分には、利用者との会

話を行うプログラム、ソフトウェア本体の実行時間等の計量を開始するために、P割込み要求テーブルに割込み要求をき込むプログラムを置くことができる。その後、ソフトウェア本体に制御を移す命令を発行する。この命令では、Pビットのクリアが行われる。

以後、P割込みとRPE命令によって、US状態とPM状態とを周を行き来しながらプログラムの実行がなされる。権利管理対の実行を終了する場合、オペレーティング・システムは、消去したい権利管理対のPMIDをパラメータとするPMT消去命令を発行する。これにより、ハードウェアによって、権利管理テーブルの対応するレコードおよびPITの該当エントリが消去される。

権利管理に関する命令について、例えばいかなる命令を用意すればよいかについては、以上の実施例の説明で明らかであるが、各命令の機能変更、拡張は任意になし得る。

(発明の効果)

- 31 -

以上説明したように、本発明によれば、マルチプログラミングの環境において、保水性、柔軟性に優れた権利管理を行うことが可能になり、ソフトウェアを自由に大量に流通させ、かつソフトウェアの保護が充分であるシステムを構築できるようになる。

#### 4. 図面の簡単な説明

第1図は本発明の原理説明図。

第2図は本発明を用いたソフトウェアの流通形態の例を説明するための図。

第3図は本発明による権利管理の説明図。

第4図は権利管理状態の状態遷移図。

第5図は本発明に用いるハードウェア構成例。

第6図は権利管理対象となるソフトウェアの構成例。

第7図は権利管理テーブルの構成例。

第8図はP割込み要求テーブル構成例。

第9図はS割込み制御の例。

第10図はS割込みからの復帰制御の例。

- 33 -

- 32 -

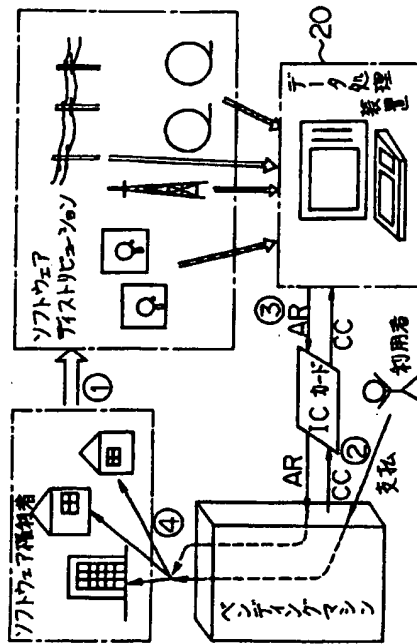
第11図は本発明の一実施例におけるスタックの使用例。

第12図は本発明の一実施例における空間と態の関係説明図を示す。

図中、10は権利管理対象プログラム、11は許諾条件プログラム、12はソフトウェア本体、13は権利管理割込み発生機構、14は権利管理状態の変更処理、15はプログラムステークスワード、16は権利管理ID、17は権利管理テーブル、18は復号化回路を表す。

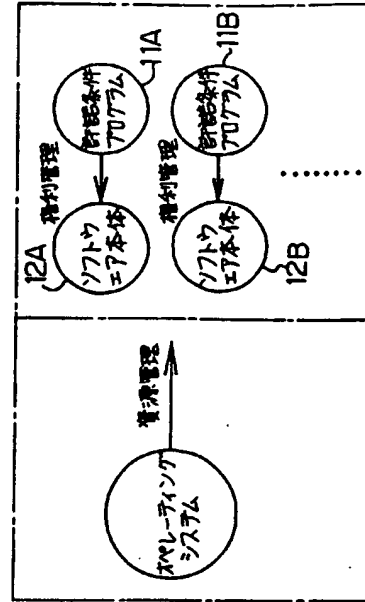
特許出願人 森 亮一

代理人 弁理士 小笠原 吉義 (外2名)



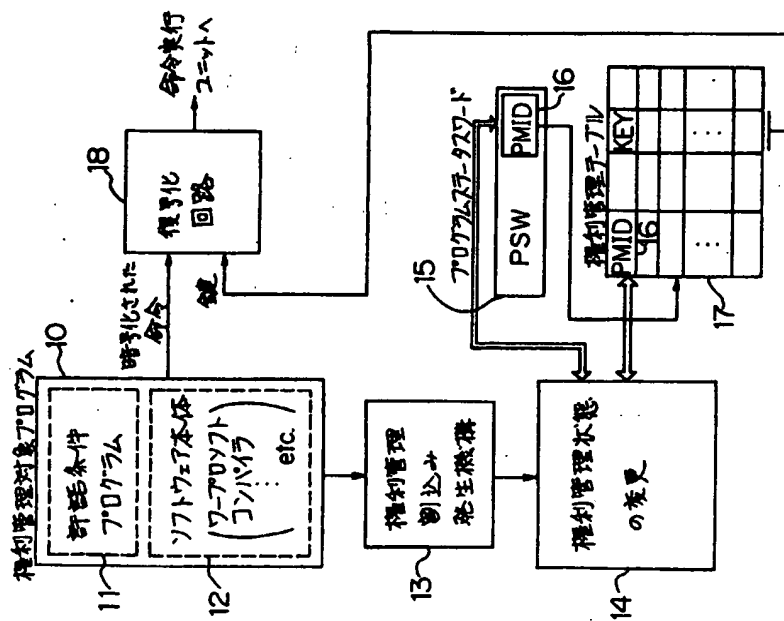
本発明を用いたソフトウェア流通形態

第2図



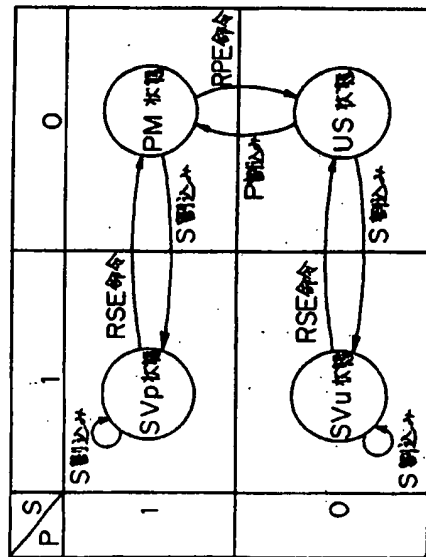
本発明による権利管理

第3図

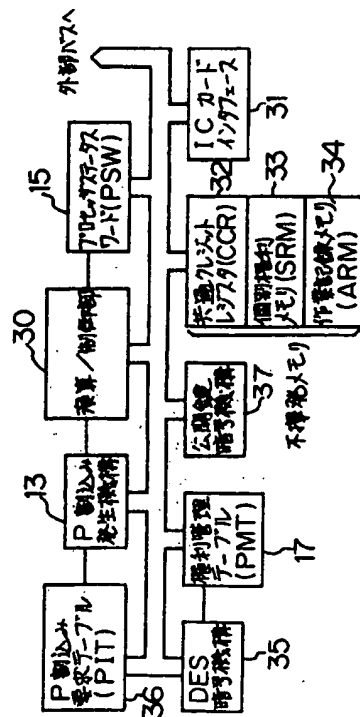


本発明の原理説明図

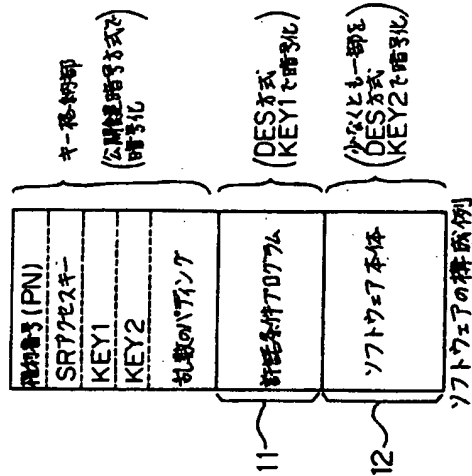
第1図



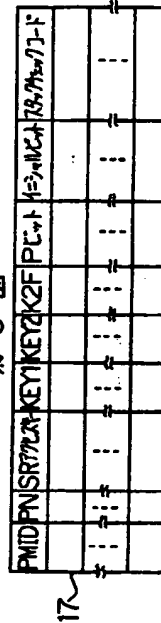
本発明の構成例  
第4図



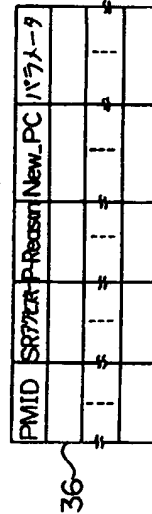
本発明に用いるハードウェア構成例  
第5図



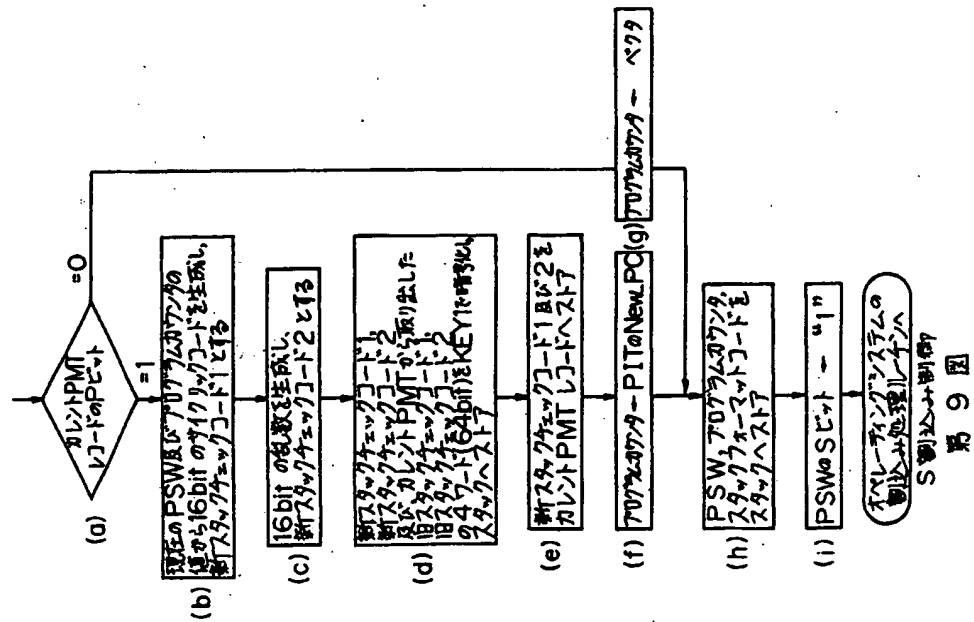
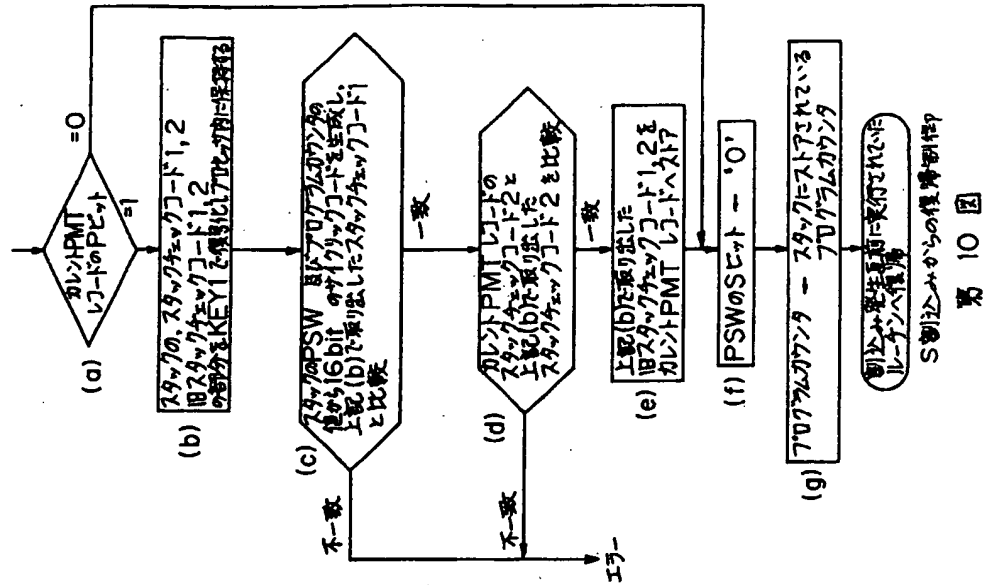
ソフトウェアの構成例  
第6図



権利管理テーブルの構成例  
第7図



P-Moduleの構成例  
第8図



15	0
P S W	
プログラムカウンタ上位	
プログラムカウンタ下位	
スタックフォーマットコード	
スタックチェックコード1	
スタックチェックコード2	
旧スタックチェックコード1	
旧スタックチェックコード2	

15	0
P S W	
プログラムカウンタ上位	
プログラムカウンタ下位	
スタックフォーマットコード	

(b) US状態時におけるS割込み発生時

(a) PM状態時におけるS割込み発生時

15	0
P S W	
プログラムカウンタ上位	
プログラムカウンタ下位	
スタックフォーマットコード	

15	0
P S W	
プログラムカウンタ上位	
プログラムカウンタ下位	
スタックフォーマットコード	

(c) SV状態時におけるS割込み発生時 (d) US状態時におけるP割込み発生時  
スタックの使用例

第 11 図

	SV状態	PM状態	US状態
空間	スーパーバイザ空間	ユーザ空間	
鍵	—	カレントPMTのKEY1	カレントPMTのKEY2

空間と鍵の関係  
第 12 図